UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/821,085	04/08/2004	Cindy Howard	SBS001/150068	1137

23444 7590 03/04/2008
ANDREWS & KURTH, L.L.P.
600 TRAVIS, SUITE 4200
HOUSTON, TX 77002

EXAMINER	
WANG, JUE S	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
03/04/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/821,085	**Applicant(s)** HOWARD ET AL.	
	Examiner JUE S. WANG	**Art Unit** 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 December 2007*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *2-4,6,8,9,11,12,16,18-20 and 22-36* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *2-4,6,8,9,11,12,16,18-20 and 22-36* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *08 April 2004* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. Claims 2-4, 6, 8, 9, 11, 12, 16, 18-20, 22-36 have been examined.

2. Claims 1, 5, 7, 10, 13-15, 17, and 21 were cancelled in amendment dated 12/21/2007.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 30-35 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

A. The following claim language is indefinite and not clear:

i. As per Claims 30-34, the term “the first data file” is used. While it is apparent that this term refers to the “fist non-source-code data file”, applicants are required to be consistent in their claim language and “the first data file” should be amended to read “the first non-source-code data file”.

Appropriate corrections are required.

Any claim not specifically addressed, above, is being rejected as incorporating the deficiencies of a claim upon which it depends.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 2, 9, 11, 12, 16, 18, 23, 24, 28, 29, and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Abrams (US 6,151,608), in view of Chang et al., (US 6,308,178 B1, hereinafter Chang), further in view of Rechter (US 6,698,014 B1).

7. As per claim 28, Abrams teaches the invention as claimed, including a method for converting a source legacy database system to a target relational database system (see Fig 3, column 8, lines 37-59, column 11, lines 46-56), the method comprising the steps of:

collecting the entirety of said source legacy database (see column 6, lines 16-27, column 8, line 50, column 12, lines 5-19);

storing said entirety in a relational common construct database (see column 12, lines 5-19);

resolving said entirety into a plurality of basic constituent elements thereof (see column 6, lines 16-27, column 7, lines 23-24, column 12, lines 5-19, i.e., the data in the database are considered as basic constituent elements);

storing said plurality of basic constituent elements in said relational common construct database (see column 6, lines 16-27, column 7, lines 23-24, column 12, lines 5-19);

writing at least one template in an algorithmic language (see column 12, lines 35-45);

writing said target relational database from said at least one template as a function of said plurality of basic constituent elements, said at least one template controlling said writing (see Fig 4, column 15, lines 5-20).

Abrams does not explicitly teach that the template is stored in the relational common construct database.

Chang teaches a system for data migration including a knowledge repository that stores the migration data and the rules for migrating the data (see Figs 1, 2, 4, column 4, lines 17-37, column 6, lines 29-38).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Abrams such that the templates are stored in the common construct database because it provides a repository of information to be used by the migration component that may be expanded to define migrations to different or updated target applications (see column 2, lines 42-60, column 3, lines 11-23 of Cheng).

Abrams also does not explicitly teach that the source legacy database system defines an environment that includes a first non-source-code data file and a first computer program source code file.

Rechter teaches a migration program that converts source code and environment data from a first platform to a second platform, including resolving the source code into a plurality of basic constituent elements and storing the plurality of basic constituent elements in a database (see Fig 1, column 4, line 15 – column 5, line 15).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Abrams to define an environment that includes a first non-source-code data file and a first computer program source code file in the database as taught by Rechter because it allows conversion of source code form a first software language and platform to a second software language and platform (see column 2, lines 26-28 of Rechter) and the migration system of Abrams can accommodate this additional migration data because it can be used to support virtually any function which involves translation, transformation, or migration of data from a source to an Oracle destination including migrating data from development to test, production environments and consolidating data from different sources into a single set of Oracle tables (see column 17, lines 19-25, 44-54, column 19, lines 20-23 of Abrams).

8. As per claim 2, Abrams further teaches writing a condition statement in said at least one template, said condition statement having a condition expression contained therein, evaluating said condition expression, performing a first task if said condition expression is evaluated to be a first value, and performing a second task if said condition expression is evaluated to be a second value (see column 5, lines 25-28, column 14, lines 50-58).

9. As per claim 29, Abrams teaches the invention as claimed, including a system for converting a source database system to a target database system, said target database having substantially equivalent functionality as said source database (see Fig 3, column 8, lines 37-59, column 11, lines 46-56), the system comprising:

at least one computer system (column 6, lines 1-6, column 11, lines 46-56);

a relational database management system structured for execution by said at least on computer system (see column 5, lines 46-51, column 6, lines 16-20);

a deconstruction program structured for execution by said at least one computer system (i.e., the data map architect, see Fig 3, column 9, lines 35-40, column 11, line 64 - column 12, line 4);

a construction program structured for execution by said at least one computer system (i.e., the update processor, see Fig 3, column 12, line 46 – column 13, line 45);

a relational common construct database operably coupled by said relational database management system to said deconstruction program and said construction program (see column 6, lines 16-20, column 12, lines 5-22, 52-55, column 13, lines 26-28; EN: the temporary table is considered a relational database);

said source database system stored in said relational common construct database (see column 6, lines 16-20, column 12, lines 5-22);

said deconstruction program operatively coupled to said source database system for resolving said source database system into a plurality of basic constituents (see column 6, lines 16-27, column 7, lines 23-24, column 12, lines 5-19, i.e., the data in the database are considered basic constituent elements);

each of said plurality of basic constituents individually stored in said relational common construct database (see column 6, lines 16-27, column 7, lines 23-24, column 12, lines 5-19);

a template written in an algorithmic language operatively coupled to said construction program for controlling said construction program in writing said target database system as a function of said plurality of basic constituents (see Fig 3, 4, column 12, lines 35-45, column 15, lines 5-20),

said construction program operatively coupled to said target database system (see column 13, lines 29-41);

Abrams does not explicitly teach that the template is stored in the relational common construct database and the target database system is stored in the relational common construct database.

Chang teaches a system for data migration including a knowledge repository that stores the migration data and the rules for migrating the data (see Figs 1, 2, 4, column 4, lines 17-37, column 6, lines 29-38).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Abrams such that the templates are stored in the common construct database because it provides a repository of information to be used by the migration component that may be expanded to define migrations to different or updated target applications (see column 2, lines 42-60, column 3, lines 11-23 of Cheng). Furthermore, it would have been obvious to store the target database in the relational common construct database as well because the target database is part of the migration engine (see Figs 3, 4, column 15, lines 25-56 of Abrams), so it would have been obvious to one of ordinary skill in the art to have included the target database in a repository of information for the migration to facilitate the audit trail maintained by the system of Abrams to reverse a upload function (see column 13, lines 41-45 of Abrams).

Abrams does not teach that the source database system composed of a plurality of basic constituents that are arranged to form at least a first non-source-code data file and a first computer program source code file.

Rechter teaches a migration program that converts source code and environment data from a first platform to a second platform, including resolving the source code into a plurality of basic constituent elements and storing the plurality of basic constituent elements in a database (see Fig 1, column 4, line 15 – column 5, line 15).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Abrams to define an environment that includes a first non-source-code data file and a first computer program source code file in the database as taught by Rechter because it allows conversion of source code form a first software language and platform to a second software language and platform (see column 2, lines 26-28 of Rechter) and the migration system of Abrams can accommodate this additional migration data because it can be used to support virtually any function which involves translation, transformation, or migration of data from a source to an Oracle destination including migrating data from development to test, production environments and consolidating data from different sources into a single set of Oracle tables (see column 17, lines 19-25, 44-54, column 19, lines 20-23 of Abrams).

10. As per claim 9, Abrams does not teach that the deconstruction includes at least one language dependent parser.

Rechter teaches that the deconstruction includes at least one language dependent parser (see column 4, lines 60-67).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Abrams to include at least one language dependent parser as taught by Rechter because it is part of the conversion process of source code form a first software language and platform to a second software language and platform (see column 2, lines 26-28 of Rechter).

7. As per claim 11, Abrams does not teach an original environment description of the database system stored in the relational common construct database.

Rechter teaches storing in a database an original environment description for the source code being converted (see column 3, lines 35-38, column 4, lines 50-59, column 6, lines 49-59).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Abrams such that an original environment description is stored in the database as taught by Rechter because it is necessary to know the supplier, version, and release of the operating system, development environments, databases, and software language of the source and target environments to convert source code from a first programming language and platform to a second programming language and platform (see column 2, lines 25-38 and column 50-59 of Rechter).

11. As per claim 12, Abrams does not teach a target environment description of the database system stored in the relational common construct database.

Rechter teaches storing in a database an original environment description for the source code being converted (see column 3, lines 35-38, column 4, lines 50-59, column 6, lines 49-59).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Fuchs such that a target environment description is stored in the database as taught by Rechter because it is necessary to know the supplier, version, and release of the operating system, development environments, databases, and software language of the source and target environments to convert source code from a first programming language and platform to a second programming language and platform (see column 2, lines 25-38 and column 50-59 of Rechter).

12. As per claim 16, Abrams teaches the construction program includes a template workbench designed and arranged for editing said template (see column 12, lines 35-39).

13. As per claim 18, Abrams further teaches the algorithmic language includes a condition statement in said at least one template, said condition statement having a condition expression contained therein, evaluating said condition expression, performing a first task if said condition expression is evaluated to be a first value, and performing a second task if said condition expression is evaluated to be a second value (see column 5, lines 25-28, column 14, lines 50-58).

14. As per claim 23, Abrams further teaches said template generally defines the structure of a target computer program source code file of said target database system, said construction program is arranged and designed to interpret said template and write said target computer program source code file therefrom (see column 12, lines 35-45, column 15, lines 33-47, column 18, lines 42-56).

15. As per claim 24, Abrams further teaches the algorithmic language includes a conditional evaluation means to control said writing in a first manner if said condition expression is evaluated to be a first value and control said writing in a second manner if said condition expression is evaluated to a second value (see column 5, lines 25-28, column 14, lines 50-58).

16. As per claim 36, Abrams further teaches said source database system is a legacy database system, and said target database system is a relational database system (see column 6, lines 62-63, column 8, line 50).

17. Claims 3, 4, 6, 19, 20, 22, 25-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Abrams (US 6,151,608), in view of Chang et al., (US 6,308,178 B1, hereinafter Chang), further in view of Rechter (US 6,698,014 B1), as applied to claims 28 and 29 above, further in view of Uner (WO 02/35349 A1).

8. As per claim 3, Abrams, Chang, and Rechter do not teach writing an iteration statement in said at least one template, said iteration statement having an iteration expression contained therein, iteratively evaluating said iteration expression, and performing a task while said iteration expression is evaluated to be a particular value.

Uner teaches a high level template language for translating data streams which contains an iteration statement (see page 13, lines 1-6 and lines 12-16). While Uner does not explicitly disclose that the template contains an iteration statement, it is obvious that the template can contain an iteration statement since the template language used to describe the template provides a means for constructing iteration statements.

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified the template language of Abrams to contain an iteration statements as taught by Uner because it provides a more expressive programming construct to specify the template.

18. As per claim 4, Abrams, Chang, and Rechter do not teach writing a subroutine statement in said template, said subroutine statement having a subroutine name contained therein, designating a portion of said at least one template by said subroutine name, and interpreting said subroutine statement as a directive to control said-writing by said portion.

Uner teaches a high level template language for translating data streams (see page 13, lines 1-6 and lines 12-16). While Uner does not explicitly state as such, it is well known in the art that high level programming languages provide a means to construct subroutines. While Uner does not explicitly disclose that the template contains a subroutine statement, it is obvious that the template can contain a subroutine statement since the template language used to describe the template provides a means for constructing subroutine statements.

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified the template language of Abrams to contain a subroutine statements as taught by Uner because it provides a more expressive programming construct to specify the template.

19. As per claim 6, Abrams, Chang, and Rechter do not teach writing a set statement in said at least one template, said set statement having a set expression and a set variable contained therein, assigning said set expression to said set variable.

Uner teaches a high level template language for translating data streams (see page 13, lines 1-6 and lines 12-16). While Uner does not explicitly state as such, it is well know in the art that any high level programming language would contain expressions for variable assignment. While Uner does not explicitly disclose that the template contains a set expression, it is obvious that the template can contain a set expression since the template language used to describe the template provides a means for constructing set expression.

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified the template language of Abrams to contain set statements as taught by Uner because it provides a more expressive programming construct to specify the template.

20. As per claims 19, 20, and 22, the limitations of these claims are substantially similar to claims 3, 4, and 6. Therefore, they are rejected using the same reasons as claims 3, 4, and 6.

21. As per claims 25-27, the limitations of these claims are substantially similar to claims 3, 4, and 6. Therefore, they are rejected using the same reasons as claims 3, 4, and 6.

22. Claim 8 is rejected under 35 U.S.C. 103(a) as being unpatentable over Abrams (US 6,151,608), in view of Chang et al., (US 6,308,178 B1, hereinafter Chang), further in view of

Rechter (US 6,698,014 B1), as applied to claims 29 above, further in view of Venter, (US 7,219,338 B2,).

9. As per claim 8, Abrams, Chang, and Rechter do not teach that the deconstruction program includes a language-determination parser.

Venter teaches a method for multi-language compilation (see abstract), including a language-determination parser (see Fig 2 and column 5, lines 62-65).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Venter to contain a language-determination parser as taught by Venter because the source code conversion program of Rechter can go from any input language to any output language (see column 10, lines 25-34 of Rechter) and it is well known in the art that different programming languages need different parsers, so the language-determination parser would provide support for determining which parser to use for the given input language.

23. Claims 30-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Abrams (US 6,151,608), in view of Chang et al., (US 6,308,178 B1, hereinafter Chang), further in view of Chen et al., (US 003/0188036 A1, hereinafter Chen).

24. As per claim 30, Abrams teaches the invention as claimed, including a system for converting a source database system to a target database system (see Fig 3, column 8, lines 37-59, column 11, lines 46-56), the system comprising:

at least one computer system having a relational database management system and a common construct database (see column 5, lines 46-51, column 6, lines 1-6, 16-20, column 11, lines 46-56, column 12, lines 5-22, 52-55, column 13, lines 26-28);

storing the database in said common construct database by said relational database management system of said at least one computer system (see column 6, lines 16-20, column 12, lines 5-22);

a conversion software program resident on said at least one computer system designed and arranged to receive data in the database to produce as an output said target database system (see Fig 3, column 15, lines 25-61);

a template written in an algorithmic language that controls said conversion software program and defines the structure of said target database system (see Fig 3, 4, column 12, lines 35-45, column 15, lines 5-20).

Abrams does not teach that the template is store in the common construct database.

Chang teaches a system for data migration including a knowledge repository that stores the migration data and the rules for migrating the data (see Figs 1, 2, 4, column 4, lines 17-37, column 6, lines 29-38).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Abrams such that the templates are stored in the common construct database because it provides a repository of information to be used by the migration component that may be expanded to define migrations to different or updated target applications (see column 2, lines 42-60, column 3, lines 11-23 of Cheng).

Abrams does not teach the source database defining an environment that includes a first non-source-code data file and a first computer program source code file, that the first data file and the first software program source code are input into a conversion software program to produce the target database system.

Chen teaches a system for program migration that converts source code and non source code files from a first platform to a second platform (see Fig 2, [0010], [0035]-[0039]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Abrams to define an environment that includes a first non-source-code data file and a first computer program source code file in the database as taught by Chen because it allows program migration from a source server environment to a destinations server environment (see [0010] of Chen) and the migration system of Abrams can accommodate this additional migration data because it can be used to support virtually any function which involves translation, transformation, or migration of data from a source to an Oracle destination including migrating data from development to test, production environments and consolidating data from different sources into a single set of Oracle tables (see column 17, lines 19-25, 44-54, column 19, lines 20-23 of Abrams).

25. As per claim 31, Chen further teaches the first data file is structured as a software report (see [0036]).

26. As per claim 32, Chen further the first data file is structured as a utility (see [0038], [0039], [0042]).

27. As per claim 33, Chen further the first data file is structured as a descriptor ([0037]).

28. As per claim 34, Chen further teaches the first data file is structured as a repository extract (see [0035]).

29. As per claim 35, Abrams further teaches said source database system is a legacy database system, and said target database system is a relational database system (see column 6, lines 62-63, column 8, line 50).

Response to Arguments

30. Rejection of claims 28-30:

31. Applicants' arguments with respect to claims 28-30 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

32. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Barnishan (US 6,654,950 B1) is cited to teach a software rehosting system.

33. Applicants' amendment necessitated the new ground(s) of rejection presented in this office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP §706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

34. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jue S. Wang whose telephone number is (571) 270-1655. The examiner can normally be reached on M-Th 7:30 am - 5:00pm (EST).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on 571-272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Jue Wang
Examiner
Art Unit 2193

/Lewis A. Bullock, Jr./

Supervisory Patent Examiner, Art Unit 2193